

# Hybrid Monte Carlo algorithm for lattice QCD with two flavors of dynamical Ginsparg-Wilson quarks

CHUAN LIU

Department of Physics  
Peking University  
Beijing 100871 P. R. China

February 1, 2008

## Abstract

We study aspects concerning numerical simulations of Lattice QCD with two flavors of dynamical Ginsparg-Wilson quarks with degenerate masses. A Hybrid Monte Carlo algorithm is described and the formula for the fermionic force is derived for two specific implementations. The implementation with optimal rational approximation method is favored both in CPU time and memory consumption.

# 1 Introduction

Recently, in a series of publications[1, 2, 3, 4, 5, 6], it has become clear that, if one would modify the chiral transformation laws away from their canonical form in the continuum, chiral symmetries can be preserved on the lattice without the problems of fermion doubling. The lattice Dirac operator  $D$  for these fermions satisfies the Ginsparg-Wilson [7] relation<sup>1</sup>,

$$\gamma_5 D + D \gamma_5 = a D \gamma_5 D \quad , \quad (1)$$

where  $a$  is the lattice spacing. As a consequence of the Ginsparg-Wilson relation (1), it is easy to show that the fermion action,

$$S_F = \sum_{x,y} \bar{\psi}(x) D_{x,y} \psi(x) \quad , \quad (2)$$

is invariant under lattice chiral transformations, and chiral symmetry will protect the quark masses away from the additive renormalizations.

The chiral properties of the Ginsparg-Wilson fermions is a direct result of the Ginsparg-Wilson relation. In particular, several types of fermion actions could be written down which all fulfill the condition. For definiteness, one particular choice [8, 5] is adopted in this paper, namely:

$$\begin{aligned} aD &= 1 - H_W (H_W^\dagger H_W)^{-1/2} \quad , \\ H_W &= 1 + s - a \sum_{\mu} \frac{1}{2} [\gamma_{\mu} (\nabla_{\mu} + \nabla_{\mu}^*) - a \nabla_{\mu}^* \nabla_{\mu}] \quad , \\ &= (-3 + s) \delta_{x,y} + \frac{1}{2} \sum_{\mu} [(1 - \gamma_{\mu}) U_{\mu}(x) \delta_{x+\mu,y} + (1 + \gamma_{\mu}) U_{\mu}^\dagger(x - \mu) \delta_{x-\mu,y}] \end{aligned} \quad (3)$$

where  $s$  is a parameter satisfying  $|s| < 1$ . The lattice covariant derivatives  $\nabla_{\mu}^*$  and  $\nabla_{\mu}$  are defined as usual,

$$\begin{aligned} \nabla_{\mu} \psi &= U_{\mu}(x) \psi(x + \mu) - \psi(x) \quad , \\ \nabla_{\mu}^* \psi &= \psi(x) - U_{\mu}^\dagger(x - \mu) \psi(x) \quad . \end{aligned} \quad (4)$$

Due to their decent chiral properties, it is quite tempting to investigate the possibility of performing Monte Carlo simulations using Ginsparg-Wilson fermions, replacing the conventional Wilson fermions. Albeit their seemingly non-local appearance, the fermion matrix (3) is in fact local (with exponentially decaying tails), as long as the parameter  $s$  in the matrix is chosen appropriately [9, 10]. The locality property of the fermion matrix will enable us to use iterative methods in Krylov space whenever inversion of the matrix becomes necessary.

When performing the inversion of the fermion matrix (3), one would encounter the problem of yet another matrix inversion of a fractional power. Recently, proposals have been put forward

---

<sup>1</sup>They are therefore named Ginsparg-Wilson fermions.

[8, 12] which make such inversions possible. Some numerical calculations in quenched QCD [11] for Ginsparg-Wilson fermions already indicate that these fermions indeed have anticipated chiral properties. However, it is also realized that quenched calculations with Ginsparg-Wilson fermions are more costly than with conventional Wilson fermions primarily due to fractional inversion of the matrix  $H_W^\dagger H_W$ . Two specific types of methods will be discussed in this paper. One is the method proposed by Bunk [12], which will be called fractional inverter method, or FIM. The other method is the optimal rational approximation method, or ORAM, proposed in Ref.[11]. It was reported in Ref. [11] that ORAM converges faster than FIM for a desired accuracy and a given condition number of the matrix  $H_W^\dagger H_W$ . Now, each multiplication with the fermion matrix  $D$  for Ginsparg-Wilson fermions is equivalent to  $2N + 1$  multiplications with matrix  $H_W^\dagger$  or  $H_W$ , where  $N$  is some integer. With FIM,  $N$  is the number of highest order Legendre polynomial kept in the iteration procedure. With ORAM,  $N = N_{cg}$  is the number of conjugate gradient <sup>2</sup> iterations needed to perform the multi-shift matrix inversions. This number is determined by the condition number of the matrix  $H_W^\dagger H_W$  and the accuracy desired. Therefore, the calculations with Ginsparg-Wilson fermions is at least more costly than conventional Wilson fermions by a factor of  $2N + 1$ .

Although it is already quite costly in the quenched case, it remains an tempting problem to simulate *dynamical* Ginsparg-Wilson fermions. No algorithms including dynamical fermions have been tested on these newly proposed fermions. In this paper, it is shown that a Hybrid Monte Carlo algorithm would do the job, however, just as in the quenched case, the simulation is more costly than simulating *dynamical* Wilson fermions. Also, in the calculation of the fermionic force to gauge links, using two different methods for the fractional inversion results in very different memory and CPU time consumptions. For FIM, it seems that  $O(N)$  pseudofermion fields have to be stored in order to make the simulation tractable. For ORAM, only a moderate number of pseudofermion fields have to be stored, and the number of matrix multiplications also increases slower than in FIM. In this paper, we will concentrate on a Hybrid Monte Carlo algorithm for simulation of dynamical Ginsparg-Wilson fermions. The fermionic force for the gauge links, which is the crucial part for the dynamical fermion simulation, will be calculated for both ORAM and FIM. General properties of the two methods are compared. Test runs on small lattices with gauge group  $SU(3)$  are now being investigated [13] and detailed results will be reported later.

This paper is organized in the following manner. In Section 2, the Hybrid Monte Carlo algorithm suitable for simulating dynamical Ginsparg-Wilson fermions are described and the formula for the force of the gauge link is derived, in both ORAM and FIM. These two methods are compared in the dynamical simulation in terms of CPU time consumption and memory consumption. Possible improvement methods are also addressed. Some concluding remarks are

---

<sup>2</sup> By the phrase "conjugate gradient", we mean all possible iterative algorithms in Krylov space: conjugate gradient, minimal residue, Bi-conjugate gradient, etc.

in Section 3.

## 2 The Hybrid Monte Carlo algorithm

The basic formalism of Hybrid Monte Carlo algorithm [14] remains the same as in the conventional Wilson case [15]. Only the fermionic force has to be re-derived for the Ginsparg-Wilson case, which will be dealt with below in detail. The effective action with the pseudofermion contribution now reads:

$$S_{eff} = S_g[U_\mu(x)] + \phi^\dagger Q^{-2} \phi , \quad (5)$$

where the fermion matrix  $Q \equiv \gamma_5(D + m)$  is hermitian and  $\phi(x)$  being the pseudofermion field generated at the beginning of a Hybrid Monte Carlo trajectory from Gaussian noise. We have also assumed that two flavors of quarks have degenerate masses. At each molecular dynamics step in a Hybrid Monte Carlo trajectory, one has to find solution vectors  $X_1 \equiv Q^{-2}\phi$  and  $X_2 \equiv Q^{-1}\phi = QX_1$  from an iterative algorithm (conjugate gradient, for example) in Krylov space. Then, the total force with respect to the gauge fields can be found by investigating the variation of the action under infinitesimal changes of the gauge fields:

$$\begin{aligned} \delta S_{eff} &= \sum_{x,\mu} Tr[V_\mu(x)\delta U_\mu(x) + h.c.] + \delta S_f , \\ \delta S_f &= \delta[\phi^\dagger Q^{-2}\phi] = Tr[F_\mu(x)\delta U_\mu(x) + h.c.] . \end{aligned} \quad (6)$$

The gauge staple  $V_\mu(x)$  comes solely from the pure gauge part of the action and could be obtained with little cost. The fermionic forces  $F_\mu(x)$ , however, is much more costly. Once the fermionic forces are obtained, the standard Hybrid Monte Carlo updating procedure can be carried on just as in the conventional Wilson case.

To derive the formula for the fermionic force, we take the variation of the fermionic part of the action and get,

$$\delta S_f = X_1^\dagger (-\delta Q) X_2 + X_2^\dagger (-\delta Q) X_1 \quad (7)$$

The variation of the matrix  $Q$  contains two parts, one being simple, namely

$$-\delta_1 Q = \gamma_5(\delta H_W)(H_W^\dagger H_W)^{-1/2} , \quad (8)$$

the other being quite complicated, i.e.

$$-\delta_2 Q = \gamma_5 H_W \delta(H_W^\dagger H_W)^{-1/2} , \quad (9)$$

which depends on the detailed implementation of the fractional inversion of the matrix. We now proceed to discuss the fermionic forces in ORAM and FIM respectively.

## 2.1 Fermionic force in ORAM

We first present the force in ORAM which is more straightforward. We recall that, this approximation amounts to approximate the function  $z(z^2)^{-1/2}$  in the interval  $[0, 1]$  by a ratio of two polynomials:

$$z(z^2)^{-1/2} = z \left( c_0 + \sum_{k=1}^N \frac{c_k}{z^2 + q_k} \right) , \quad (10)$$

It is an approximation similar to the Padé approximation used in Ref.[16]. For details about this approximation and how to get coefficients  $c_k$ , consult [11] and references therein. Applying this method to the hermitian matrix  $\gamma_5 H_W$ , we immediately obtain the following expression for the variation of the fermionic action:

$$\begin{aligned} \delta S_f &= c_0 \text{Tr}(X_2 \otimes X_1^\dagger \gamma_5 \delta H_W) + \sum_{k=1}^N c_k \text{Tr}(\zeta_{2,k} \otimes X_1^\dagger \gamma_5 \delta H_W) \\ &+ c_0 \text{Tr}(X_1 \otimes X_2^\dagger \gamma_5 \delta H_W) + \sum_{k=1}^N c_k \text{Tr}(\zeta_{1,k} \otimes X_2^\dagger \gamma_5 \delta H_W) \\ &- \sum_{k=1}^N c_k \text{Tr} \left( \zeta_{2,k} \otimes \xi_{1,k}^\dagger H_W^\dagger \delta H_W + \xi_{2,k} \otimes \xi_{1,k}^\dagger \gamma_5 \delta H_W \right) , \\ &- \sum_{k=1}^N c_k \text{Tr} \left( \zeta_{1,k} \otimes \xi_{2,k}^\dagger H_W^\dagger \delta H_W + \xi_{1,k} \otimes \xi_{2,k}^\dagger \gamma_5 \delta H_W \right) , \\ \zeta_{i,k} &= \frac{1}{(\gamma_5 H_W)^2 + q_k} X_i , \quad \xi_{i,k} = \gamma_5 H_W \frac{1}{(\gamma_5 H_W)^2 + q_k} X_i . \end{aligned} \quad (11)$$

In the above formula, “ $Tr$ ” stands for taking trace in both Dirac and color space and a summation over the whole lattice points. The symbol  $\otimes$  stands for direct product of two vectors in color space. Therefore, the fermionic force is obtained as

$$\begin{aligned} F_\mu(x) &= \frac{c_0}{2} \text{tr}_{Dirac} [X_2(x+\mu) \otimes X_1^\dagger(x) \gamma_5 (1 - \gamma_\mu) + X_1(x+\mu) \otimes X_2^\dagger(x) \gamma_5 (1 - \gamma_\mu)] \\ &+ \sum_{k=1}^N \text{tr}_{Dirac} \frac{c_k}{2} [\zeta_{2,k}(x+\mu) \otimes X_1^\dagger(x) \gamma_5 (1 - \gamma_\mu)] \\ &+ \sum_{k=1}^N \text{tr}_{Dirac} \frac{c_k}{2} [\zeta_{1,k}(x+\mu) \otimes X_2^\dagger(x) \gamma_5 (1 - \gamma_\mu)] \\ &- \sum_{k=1}^N \text{tr}_{Dirac} \frac{c_k}{2} [\zeta_{2,k}(x+\mu) \otimes [H_W \xi_{1,k}]^\dagger(x) (1 - \gamma_\mu)] \\ &- \sum_{k=1}^N \text{tr}_{Dirac} \frac{c_k}{2} [\xi_{2,k}(x+\mu) \otimes \xi_{1,k}^\dagger(x) \gamma_5 (1 - \gamma_\mu)] \end{aligned}$$

$$\begin{aligned}
& - \sum_{k=1}^N tr_{Dirac} \frac{c_k}{2} [\zeta_{1,k}(x+\mu) \otimes [H_W \xi_{2,k}]^\dagger(x)(1-\gamma_\mu)] \\
& - \sum_{k=1}^N tr_{Dirac} \frac{c_k}{2} [\xi_{1,k}(x+\mu) \otimes \xi_{2,k}^\dagger(x) \gamma_5(1-\gamma_\mu)] ,
\end{aligned} \tag{12}$$

where the trace  $tr_{Dirac}$  is only taken within the Dirac space.

## 2.2 Fermionic force in FIM

In FIM, we would like to solve for  $\xi$  satisfying:

$$M^{1/2} \xi = X \quad , \text{ given } X. \tag{13}$$

where the matrix  $M \equiv H_W^\dagger H_W$  by setting:

$$M = c(1 + t^2 - 2tA) \quad , \tag{14}$$

with the parameters  $c$  and  $t$  chosen in such a way that all eigenvalues of the matrix  $A$  lie within  $[-1, 1]$ . To be more specific, we choose,

$$t = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \quad , \quad c = \frac{(\sqrt{\kappa} + 1)^2}{4\lambda_{min}} \quad , \tag{15}$$

where  $\lambda_{min}$  ( $\lambda_{max}$ ) is the lowest (highest) eigenvalue of the matrix  $H_W^\dagger H_W$  and  $\kappa \equiv \lambda_{max}/\lambda_{min}$  is the condition number. Then, the solution to eq. (13) may be written as:

$$\xi = c^{-1/2} \sum_{k=0}^{\infty} t^k P_k(A) \cdot X = \sum_{k=0}^{\infty} s_k \quad , \tag{16}$$

where  $P_k(z)$  are Legendre polynomials. Therefore, an approximant for the solution at the  $n$ -th level is

$$\xi_n = \sum_{k=0}^n s_k \quad . \tag{17}$$

The shifts <sup>3</sup>,  $s_k$ , defined as

$$s_k = c^{-1/2} t^k P_k(A) \cdot X \quad , \tag{18}$$

satisfy the following recursion relations:

$$\begin{aligned}
s_{-1} &= 0 \quad , s_0 = c^{-1/2} X \quad , \\
s_k &= (2 - 1/k) t A s_{k-1} - (1 - 1/k) t^2 s_{k-2} \quad .
\end{aligned} \tag{19}$$

---

<sup>3</sup>A extra factor  $t^k$  has been included in the definition of  $s_k$  as compared with Ref.[12]

For the case of Legendre polynomials, it is claimed that the following bound for the residue is obtained [12]:

$$|\xi - \xi_n|/|\xi| = |R_n(A)| \leq t^{n+1} = \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^{n+1} , \quad (20)$$

which asserts the exponential convergence of the iteration.

For the vectors  $\delta(H_W^\dagger H_W)^{-1/2} X_i$ , similar strategy could be applied,

$$\delta(M^{-1/2})\eta = c^{-1/2} \sum_{n=0}^{N_{cut}} t^n \delta P_n(A) \eta , \quad \text{given } \eta, \quad (21)$$

where  $\eta$  represents either  $X_1$  or  $X_2$  and  $N_{cut}$  is the highest order of Legendre polynomials kept in the approximation. In an analogous manner, we define,

$$\delta_k \equiv c^{-1/2} t^k (\delta P_k(A)) \cdot \eta , \quad (22)$$

which satisfy the following recursion relation:

$$\begin{aligned} (k+1)\delta_{k+1} + kt^2\delta_{k-1} &= (2k+1)t(\delta A s_k + A\delta_k) . \\ \delta_{-1} &= 0 , \delta_0 = 0 , \end{aligned} \quad (23)$$

Using this relation,  $\delta_k$  could be expressed as:

$$\delta_k = \sum_{l=0}^{k-1} t L_k^l(A) \delta A s_l . \quad (24)$$

The coefficients  $L_k^l(A)$  are polynomials in  $A$  with degree  $(k-l-1)$  and can be expressed as Legendre polynomials,

$$L_k^l(A) = \frac{2l+1}{l+1} t^{k-l-1} P_{k-l-1}\left(\frac{2l+3}{l+2} A\right) . \quad (25)$$

After rearranging the double summation and some trivial algebra, we get the following formula for the variation of the fermionic action:

$$\begin{aligned} \delta S_f &= Tr(Z_1 \otimes X_2^\dagger \gamma_5 \delta H_W) + Tr(Z_2 \otimes X_1^\dagger \gamma_5 \delta H_W) \\ &- \frac{1}{2c} \sum_{l=0}^{N_{cut}} \left(\frac{2l+1}{l+1}\right) Tr[(H_W x_{2,l} \otimes T_{1,l}^\dagger) \gamma_5 \delta H_W^\dagger] \\ &- \frac{1}{2c} \sum_{l=0}^{N_{cut}} \left(\frac{2l+1}{l+1}\right) Tr[x_{2,l} \otimes (H_W \gamma_5 T_{1,l})^\dagger \delta H_W] \\ &- \frac{1}{2c} \sum_{l=0}^{N_{cut}} \left(\frac{2l+1}{l+1}\right) Tr[(H_W x_{1,l} \otimes T_{2,l}^\dagger) \gamma_5 \delta H_W^\dagger] \end{aligned}$$

$$\begin{aligned}
& - \frac{1}{2c} \sum_{l=0}^{N_{cut}} \left( \frac{2l+1}{l+1} \right) Tr[x_{1,l} \otimes (H_W \gamma_5 T_{2,l})^\dagger \delta H_W] , \\
Z_i &= (H_W^\dagger H_W)^{-1/2} X_i , \quad x_{i,l} = t^l (P_l(A)) X_i , \\
T_{i,l} &= \sum_{m=0}^{N-l-1} \mathcal{S}_{i,l}^{(m)} , \quad \mathcal{S}_{i,l}^{(m)} = t^m P_m \left( \left( \frac{2l+3}{l+2} \right) \gamma_5 A \gamma_5 \right) H_W X_i .
\end{aligned} \tag{26}$$

Therefore, the following expression for the fermionic force is obtained:

$$\begin{aligned}
F_\mu(x) &= tr_{Dirac} (Z_1(x+\mu) \otimes X_2^\dagger(x) \gamma_5 (1-\gamma_\mu) + (Z_2(x+\mu) \otimes X_1^\dagger(x) \gamma_5 (1-\gamma_\mu)) \\
& - \frac{1}{2c} \sum_{l=0}^N \left( \frac{2l+1}{l+1} \right) tr_{Dirac} [(H_W x_{2,l}(x+\mu) \otimes T_{1,l}(x)) \gamma_5 (1+\gamma_\mu)] \\
& - \frac{1}{2c} \sum_{l=0}^N \left( \frac{2l+1}{l+1} \right) tr_{Dirac} [(x_{2,l}(x+\mu) \otimes [H_W^\dagger T_{1,l}]^\dagger(x)) \gamma_5 (1-\gamma_\mu)] \\
& - \frac{1}{2c} \sum_{l=0}^N \left( \frac{2l+1}{l+1} \right) tr_{Dirac} [(H_W x_{1,l}(x+\mu) \otimes T_{2,l}(x)) \gamma_5 (1+\gamma_\mu)] \\
& - \frac{1}{2c} \sum_{l=0}^N \left( \frac{2l+1}{l+1} \right) tr_{Dirac} [(x_{1,l}(x+\mu) \otimes [H_W^\dagger T_{2,l}]^\dagger(x)) \gamma_5 (1-\gamma_\mu)] .
\end{aligned} \tag{27}$$

Since the CPU cost of a simulation program with dynamical fermions is dominated by the fermion matrix times vector operations. It becomes clear that the above formula for the fermionic force is not very useful from a practical point of view. The most CPU consuming part is the calculation of the vectors  $T_{i,l}(x)$ , for all values of  $l$ , each requiring an iterative procedure, i.e. the calculation of the quantities  $\mathcal{S}_{i,l}^{(m)}$ . This implies that, in order to calculate the fermionic force, the multiplication of the matrix  $H_W^\dagger H_W$  has to be performed  $O(N_{cut}^2)$  times, where  $N_{cut}$  can become large. This would make the calculation of the fermionic force too costly.

However, there is a way around this difficulty. The price to pay will be some extra storage. Note that Legendre polynomials  $P_m(\beta(l)z)$ ,  $\beta(l)$  being the constant  $(2l+1)/(l+1)$ , could be expressed as a linear combination of Legendre polynomials of lower or equal degrees with argument changed to  $z$ , i.e.

$$P_m(\beta z) = \sum_{j=0}^m \sigma_{m,j}(\beta) P_j(z) . \tag{28}$$

With this, we could express the quantities  $T_{i,l}$  in the following way:

$$\begin{aligned}
T_{i,l} &= \sum_{m=0}^{N-l-1} f_m(l, t) t^m P_m(\gamma_5 A \gamma_5) H_W X_i , \\
f_m(t, l) &= \sum_{j=m}^{N-l-1} t^{j-m} \sigma_{j,m}(\beta(l)) .
\end{aligned} \tag{29}$$



The functions  $f_m(t, l)$  are just c-numbers and can be calculated at the beginning of the simulation. Therefore, after the vectors  $X_i$  are obtained, one can calculate the vectors  $P_m(\gamma_5 A \gamma_5) H X_i$  once for all values of  $m$  and store the resulting vectors. Thus,  $T_{i,l}$  could be obtained easily without further iteration of matrix multiplications. The coefficients  $\sigma_{m,j}(l)$  satisfy the following recursion relation:

$$\sigma_{m,j} = \left(\frac{2m-1}{m}\right)\beta(l)\left[\left(\frac{j}{2j-1}\right)\sigma_{m-1,j-1} + \left(\frac{j+1}{2j+3}\right)\sigma_{m-1,j+1}\right] - \left(\frac{m-1}{m}\right)\sigma_{m-2,j} \quad , \quad (30)$$

where the subscripts  $m, j$  should satisfy  $0 \leq j \leq m$  and a zero value is understood whenever an out-of-range subscript is encountered. Together with  $\sigma_{0,0} \equiv 1$ , the above recursion relation completely determines all coefficients  $\sigma_{m,j}(\beta(l))$  and therefore the function  $f_m(t, l)$ . Now the calculation of the quantity  $T_{i,l}$  only requires a linear combination of vectors which costs little CPU time.

### 2.3 comparison of the two methods

We now compare the CPU time consumption and memory consumptions of the two methods discussed so far for the simulation of dynamical Ginsparg-Wilson fermions. As is well known, the CPU time consumption is proportional to the number of operations of the matrix  $H_W$  on vectors. For each molecular dynamics step in the Hybrid Monte Carlo, ORAM requires  $2N_{CG}(2N_{cg} + 1)$  number of matrix multiplications to obtain the solution vector  $X_1$  and  $4N_{cg} + 4N_r$  more matrix multiplications to obtain the fermionic force. Here,  $N_{CG}$  is the number of conjugate gradient iterations needed to obtain the solution  $X_1$  and  $N_{cg}$  is the number of conjugate gradient iterations needed to obtain the vector  $(H_W^\dagger H_W + q_{min})^{-1} X_i$  for the smallest shift  $q_{min}$ . Parameter  $N_r$  is the order of the polynomials in the optimal rational approximation. ORAM also requires to store  $N_r$  pseudofermion field vectors. As a comparison, FIM requires  $2N_{CG}(2N_{cut} + 1)$  number of matrix multiplications to obtain the solution vector  $X_1$  and  $12N_{cut}$  more matrix multiplications to obtain the fermionic force, where  $N_{cut}$  is the highest order of Legendre polynomials kept in the series expansion. FIM also needs to store  $2N_{cut}$  pseudofermion field vectors.

Concerning the CPU time consumption, both method are more costly than dynamical simulations with Wilson fermions by a factor of  $2N + 1$ , where  $N = N_{cg}$  for ORAM and  $N = N_{cut}$  for FIM. From the theoretical upper bound of the error, the two methods behave in a similar manner,  $N_{cg} \sim N_{cut}$ . Practically, however, according to the experience in [11],  $N_{cg}$  is usually less than  $N_{cut}$  because the theoretical bound is saturated for FIM while it is not for ORAM. Therefore, ORAM is more favorable compared with FIM when doing simulations with dynamical Ginsparg-Wilson fermions. Concerning the memory consumption,  $N_r$  is usually much less than  $N_{cut}$  which would again put ORAM in a more favorable place.

It is clear from the above discussion that, if one would like to accelerate the simulation with dynamical Ginsparg-Wilson fermions, one has to find ways to decrease  $N_{cg}$  in ORAM or  $N_{cut}$

in FIM. These two parameters are mainly determined by the condition number of the matrix  $H_W^\dagger H_W$ . Any preconditioning methods that would decrease the condition number of the matrix (while still maintaining the shifted nature of the matrix in ORAM) will bring an improvement to the simulation of dynamical Ginsparg-Wilson fermions. It should be pointed out that other improvements, for example using better integration schemes, would apply to both methods. Test runs on small lattices are now under investigation [13] where these algorithmic issues will be further studied.

### 3 Conclusions

In this paper, possibilities of simulating dynamical Ginsparg-Wilson fermions are discussed. The formula for the fermionic force is derived for two specific implementations of the algorithm, the optimal rational approximation method (ORAM) and fractional inverter method (FIM). It turns out that, in both methods, simulating dynamical Ginsparg-Wilson fermions are more costly than simulating dynamical Wilson fermions. The extra CPU time consumption mainly comes from the fractional inversion of the matrix. In quenched simulations, it has been reported [11] that ORAM performs better than FIM. In dynamical simulations, this conclusion remains, both on CPU time consumption and memory consumption. It should be emphasized that, though being more costly, the advantage of simulating dynamical Ginsparg-Wilson fermions over dynamical Wilson fermions or quenched Ginsparg-Wilson fermions would be a much better behavior towards the chiral limit. The feasibility of such simulation has been demonstrated in this paper using a Hybrid Monte Carlo algorithm.

### Acknowledgments

This work is supported by the Chinese National Science Foundation and by the Natural Science Fund from Peking University.

### References

- [1] H. Neuberger, Phys. Lett. 417B (1997) 141; *ibid* B427 (1998) 125.
- [2] P. Hasenfratz, Nucl. Phys. B63A-C (Proc. Suppl.) (1998) 53.
- [3] P. Hasenfratz V. Laliena and F. Niedermayer, Phys. Lett. B427 (1998) 125.
- [4] P. Hasenfratz, Lattice QCD without tuning, mixing and current renormalization, hep-lat/9802007.
- [5] M. Lüscher, Phys. Lett. B428 (1998) 342.

- [6] S. Chandrasekharan, hep-lat/9805015
- [7] P. Ginsparg and K. Wilson, Phys. Rev. D25 (1982) 2649.
- [8] H. Neuberger, A practical implementation of the overlap-Dirac operator, hep-lat/9806019.
- [9] P. Hernandez, K. Jansen and M. Lüscher, hep-lat/9808010.
- [10] Ting-Wai Chiu, hep-lat/9810002.
- [11] R. G. Edwards, U. M. Heller and R. Narayanan, A study of practical implementations of the overlap Dirac operator in four dimensions, hep-lat/9807017.
- [12] B. Bunk, hep-lat/9805030.
- [13] C. Liu, work in progress.
- [14] S. Duane, A. D. Kennedy, B. J. Pendelton and D. Roweth, Phys. Lett. B195 (1987) 216.
- [15] K. Jansen and C. Liu, Comput. Phys. Commun. 99 (1997) 221.
- [16] C. Thron, S. J. Dong, K. F. Liu and H. P. Ying, Phys.Rev. D57 (1998) 1642.